

Protected Mode Programming

Let MindShare Bring “Protected Mode Programming” to Life for You

This course covers the fundamentals of the x86 Protected Mode architecture including Long Mode (the 64-bit extensions). This course is the best way to become familiar with the behavior of x86-based processors. All topics are explained in lecture format first and then the students are given programming labs to reinforce the concepts and to get hands-on experience working with x86 processors at a very low level (e.g. Nehalem, Core 2, Atom, Opteron, Phenom, etc.). Topics include system registers, descriptor tables, gates, multitasking, protection levels, exceptions, debug registers, the paged memory-management system, V86 Mode, and the 64-bit extensions of the 32-bit architecture.

The lab exercises range from printing to the screen in real mode to setting up an interrupt driven, multitasking protected mode environment, with paging turned on.

You Will Learn:

- How to write and execute protected mode programs on an x86-based system
- How to read and understand protected mode programs written by others

Course Length: 5 Days

Who Should Attend?

This course is a must for processor validation engineers who will be writing low-level assembly code to validate Protected / Long Mode features and functionality of the processor. This course is also beneficial for software and hardware engineers, technicians, and others needing a fundamental working knowledge of x86 Protected Mode operation.

Course Contents:

- **Overview**
 - We present an overview of the entire protected mode architecture. We review the real mode concepts and explain why protected mode is important and useful. Topics include descriptors, GDT, LDT, IDT, TSS, task switching, V86, system instructions, 64-bit mode, and IOPL sensitive instructions.
- **Segment Descriptors**
 - We discuss segment descriptors in detail. Topics include GDT, selectors, segment descriptor fields, changing segment registers, access rules, initializing the GDTR, and segments in 64-bit mode.
- **Memory Models**
 - We illustrate the two basic memory models, flat and multi-segmented. Topics include flat model, flat model with protection, flat model with paging, and multi-segment models.
- **Call Gates**
 - We show how call gates are used to provide controlled entry to the operating system. Topics include motivation for call gates, 32-bit call gate descriptor, 64-bit call gate descriptor, parameter passing, stack switching, and "calling" a call gate.
- **Multitasking**
 - We discuss CPU integrated multitasking. Topics include CPU defined tasks, the 32-bit TSS and descriptor, the 64-bit TSS and descriptor, executing a task switch, protection rules, LDTs, nested tasks, and interrupt tasks.
- **Exceptions and Interrupts**
 - We discuss the protected mode interrupt mechanism. Topics include review of the IVT (real mode), IDT, interrupt, trap, and task gates, protection rules, protected mode exceptions, and error codes. Included are the 64-bit versions of the IDT, interrupt, and trap gates.

- **Paging**
 - We discuss paging (address translation) and how it is used for memory management. Topics include page tables, page-table directory, CR2 and CR3, access rights, the TLB, page faults, large pages, page tables related to 36-bit addressing, and the page tables used in 64-bit mode.
- **I/O**
 - We discuss how to control I/O access in a multitasking system. Topics include IOPL flag, IOPL sensitive instructions, and IO bitmap.
- **Instruction Opcode Prefixes**
 - Mode Switching-We discuss the various prefixes used to implement 16-bit, 32-bit, and 64-bit operands and addresses. Topics include the instruction prefixes-66h and 67h, and a detailed discussion of the REX prefix for 64-bit mode.
- **Mode Switching**
 - We discuss mode switching. Mode switching includes switching from real mode to legacy protected mode and back. In legacy protected mode we discuss initializing legacy paging and address extended paging. Finally we conclude with a discussion of switching to long mode from legacy protected mode and back.
- **Debugging Support**
 - We discuss built-in debugging features of the CPU. Topics include single stepping, software breakpoints, debug registers, execution and access breakpoints, and task switch breaks. Includes both 32-bit and 64-bit modes.
- **V86 Mode**
 - We discuss the importance and operation of V86 mode. Topics include reasons for V86 mode, virtualization of memory, of I/O, and of interrupts, V86 privilege rules, paging, and reflection of interrupts.

Recommended Prerequisites:

Some experience with x86 assembly language or completion of MindShare's x86 Assembly Language Programming Course, or permission of the instructor.

Course Material:

MindShare will supply a hardcopy and electronic version of the presentation slides including the lab descriptions.